

[illegible]

SE
VO

```

LL          IIIII
LL          IIIII
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LL          II
LLLLLLLLLLL IIIII
LLLLLLLLLLL IIIII
SSSSSSSSS
SSSSSSSSS
SS
SS
SS
SS
SSSSSS
SSSSSS
SS
SS
SS
SS
SSSSSSSSS
SSSSSSSSS

```

```
1 0001 0 MODULE SCHFCB (
2 0002 0 LANGUAGE (BLISS32),
3 0003 0 IDENT = 'V04-000'
4 0004 0 ) =
5 0005 1 BEGIN
6 0006 1
7 0007 1
8 0008 1
9 0009 1
10 0010 1 *
11 0011 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
12 0012 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
13 0013 1 *
14 0014 1 * ALL RIGHTS RESERVED.
15 0015 1 *
16 0016 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
17 0017 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
18 0018 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
19 0019 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
20 0020 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
21 0021 1 * TRANSFERRED.
22 0022 1 *
23 0023 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
24 0024 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
25 0025 1 * CORPORATION.
26 0026 1 *
27 0027 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
28 0028 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
29 0029 1 *
30 0030 1 *****
31 0031 1 ++
32 0032 1
33 0033 1 FACILITY: F11ACP Structure Level 1
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1
37 0037 1 This routine searches the current volume's FCB list for the
38 0038 1 FCB representing the desired file number.
39 0039 1
40 0040 1 ENVIRONMENT:
41 0041 1
42 0042 1 STARLET operating system, including privileged system services
43 0043 1 and internal exec routines.
44 0044 1
45 0045 1 --
46 0046 1
47 0047 1
48 0048 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 13-Dec-1976 15:41
49 0049 1
50 0050 1 MODIFIED BY:
51 0051 1
52 0052 1 V03-007 CDS0006 Christian D. Saether 16-Aug-1984
53 0053 1 Mark a noexec accessed fcb as stale always.
54 0054 1
55 0055 1 V03-006 CDS0005 Christian D. Saether 7-Aug-1984
56 0056 1 Modify test for directory fcb to test fcb$l_dirindx.
57 0057 1
```



```
.. 58      0058 1 | V03-005 CDS0004      Christian D. Saether      20-July-1984
.. 59      0059 1 |      Take out backlink checking as a performance help.
.. 60      0060 1 |      Unconditionally clear FCB$V_DIR when an fcb is found.
.. 61      0061 1 |
.. 62      0062 1 | V03-004 CDS0003      Christian D. Saether      19-Apr-1984
.. 63      0063 1 |      Use REFCNT instead of ACNT.
.. 64      0064 1 |
.. 65      0065 1 | V03-003 ACG0401      Andrew C. Goldstein,      12-Mar-1984 17:27
.. 66      0066 1 |      Don't acknowledge the existence of zero access FCB's
.. 67      0067 1 |
.. 68      0068 1 | V03-002 CDS0002      Christian D. Saether      30-Dec-1983
.. 69      0069 1 |      Use L_NORM Linkage and BIND_COMMON macro.
.. 70      0070 1 |
.. 71      0071 1 | V03-001 CDS0001      Christian D. Saether      12-May-1983
.. 72      0072 1 |      Lock scan routine and raise to sched ipl to do search.
.. 73      0073 1 |
.. 74      0074 1 | A0101  ACG26369      Andrew C. Goldstein,      31-Dec-1979 17:35
.. 75      0075 1 |      Fix multi-header file interlock bug
.. 76      0076 1 |
.. 77      0077 1 | A0100  ACG00001      Andrew C. Goldstein,      10-Oct-1978 20:03
.. 78      0078 1 |      Previous revision history moved to F11A.REV
.. 79      0079 1 | **
.. 80      0080 1 |
.. 81      0081 1 |
.. 82      0082 1 | LIBRARY 'SYSS$LIBRARY:LIB.L32';
.. 83      0083 1 | REQUIRE 'SRC$:FCPDEF.B32';
.. 84      1074 1 |
.. 85      1075 1 | ! Code must be locked down.
.. 86      1076 1 | !
.. 87      1077 1 |
.. 88      1078 1 | LOCK_CODE;
.. 89      1079 1 |
```

```

91 1080 1 GLOBAL ROUTINE SEARCH_FCB (FILE_ID) : L_NORM =
92 1081 1
93 1082 1 ++
94 1083 1
95 1084 1 FUNCTIONAL DESCRIPTION:
96 1085 1
97 1086 1 This routine searches the current volume's FCB list for the
98 1087 1 FCB representing the desired file number.
99 1088 1
100 1089 1
101 1090 1 CALLING SEQUENCE:
102 1091 1 SEARCH_FCB (ARG1)
103 1092 1
104 1093 1 INPUT PARAMETERS:
105 1094 1 ARG1: address of desired file ID
106 1095 1
107 1096 1 IMPLICIT INPUTS:
108 1097 1 CURRENT_VCB: VCB address of volume
109 1098 1
110 1099 1 OUTPUT PARAMETERS:
111 1100 1 NONE
112 1101 1
113 1102 1 IMPLICIT OUTPUTS:
114 1103 1 NONE
115 1104 1
116 1105 1 ROUTINE VALUE:
117 1106 1 FCB address if found
118 1107 1 zero if not
119 1108 1
120 1109 1 SIDE EFFECTS:
121 1110 1 NONE
122 1111 1
123 1112 1 --
124 1113 1
125 1114 2 BEGIN
126 1115 2
127 1116 2 MAP
128 1117 2 FILE_ID : REF BBLOCK; ! file ID arg
129 1118 2
130 1119 2 LOCAL
131 1120 2 FCB : REF BBLOCK, ! current FCB being looked at
132 1121 2 FIDNUM : WORD,
133 1122 2 FIDNMX : BYTE,
134 1123 2 CURVCB : REF BBLOCK;
135 1124 2
136 1125 2 BIND_COMMON;
137 1126 2
138 1127 2 LABEL
139 1128 2 SCAN:
140 1129 2
141 1130 2 ! Init the pointers and start scanning the FCB list, which is a double
142 1131 2 linked list. Check for consistency of pointers and the block ID for each
143 1132 2 FCB. We win when the FCB containing the desired file number is found;
144 1133 2 we lose at end of list (pointing back to the VCB). Note that we ignore
145 1134 2 FCB's with a zero access count that are not directory FCB's. These are
146 1135 2 temporary and on their way out. Any FCB on its way in that looks idle
147 1136 2 is protected by the file's synchronization lock, and we will never see it.
```



```
148 1137 2 : Raise IPL to SCHED to block other processes from changing FCB list
149 1138 2 : while we scan it.
150 1139 2 :
151 1140 2 :
152 1141 2 FIDNUM = .FILE_ID [FID$W_NUM];
153 1142 2 FIDNMX = .FILE_ID [FID$B_NMX];
154 1143 2 :
155 1144 2 SET_IPL (IPL$SCHED);
156 1145 2 :
157 1146 2 CURVCB = .CURRENT_VCB;
158 1147 2 FCB = .CURVCB [VCB$L_FCBFL];
159 1148 2 :
160 1149 2 SCAN:
161 1150 2 BEGIN
162 1151 2 :
163 1152 2 UNTIL .FCB EQL .CURVCB DO
164 1153 2 IF .FCB[FCB$B_TYPE] EQL DYN$C_FCB
165 1154 2 THEN
166 1155 2 BEGIN
167 1156 2 IF .FIDNUM EQL .FCB[FCB$W_FID_NUM]
168 1157 2 AND .FIDNMX EQL .FCB[FCB$B_FID_NMX]
169 1158 2 AND (.FCB[FCB$W_REFCNT] NEQ 0 OR (.FCB [FCB$L_DIRINDX] NEQ 0))
170 1159 2 THEN
171 1160 2 BEGIN
172 1161 2 :
173 1162 2 : FCB$V_DIR is used to indicate that an fcb with a refcnt of zero
174 1163 2 : may be tossed from the cache due to directory index cache replacement
175 1164 2 : at any time, regardless of whether a synchronization lock is held
176 1165 2 : for that file number (which should always be the case by the time
177 1166 2 : this routine is called).
178 1167 2 : By clearing the flag while at ipl$_sched, we prevent another
179 1168 2 : process from deallocating it after this time. Checks for whether
180 1169 2 : the flag should be set again and the setting thereof are also
181 1170 2 : done at ipl$_sched, so that this flag changes state atomically
182 1171 2 : with regard to other processes doing an otherwise uninterlocked
183 1172 2 : testbitsc test on it to determine whether or not to deallocate it.
184 1173 2 :
185 1174 2 :
186 1175 2 FCB [FCB$V_DIR] = 0;
187 1176 2 :
188 1177 2 : If there is an access lock for this fcb, but it is held in
189 1178 2 : nl mode, then mark the fcb stale to force rebuild of it
190 1179 2 : because we cannot get stale blocking routines in nl mode,
191 1180 2 : and must therefore always assume it is stale.
192 1181 2 :
193 1182 2 :
194 1183 2 IF .FCB [FCB$B_ACCLKMODE] EQL LCK$K_NLMODE
195 1184 2 AND .FCB [FCB$L_ACCLKID] NEQ 0
196 1185 2 THEN
197 1186 2 FCB [FCB$V_STALE] = 1;
198 1187 2 :
199 1188 2 LEAVE SCAN;
200 1189 2 END;
201 1190 2 :
202 1191 2 FCB = .FCB[FCB$L_FCBFL];
203 1192 2 END
204 1193 2 ELSE
```

```
.. 205      1194 3      BUG_CHECK (NOTFCBFCB, FATAL, 'FCB Linkage broken');
.. 206      1195      ! FCB not found, i.e., we dropped out of the loop.
.. 207      1196      !
.. 208      1197      !
.. 209      1198      FCB = 0;
.. 210      1199      END;
.. 211      1200      ! of block SCAN
.. 212      1201      !
.. 213      1202      SET_IPL (0);
.. 214      1203      ! lower ipl back to 0.
.. 215      1204      RETURN .FCB
.. 216      1205      ! return fcb (or 0 if not found).
.. 217      1206      ! end of routine SEARCH_FCB
..          1      END;
```

				.TITLE	SCHFCB		
				.IDENT	\V04-000\		
				.EXTRN	BUG\$_NOTFCBFCB		
				.PSECT	\$LOCKEDC1\$,NOWRT,2		
				.ENTRY	SEARCH_FCB, Save R2,R3		1080
	50	04	AC D0 00002	MOVL	FILE_ID, R0		1141
	53		60 B0 00006	MOVW	(R0), FIDNUM		
	52	05	A0 90 00009	MOVB	5(R0), FIDNMX		1142
	12		03 DA 0000D	MTPR	#3, #18		1144
	51	98	AA D0 00010	MOVL	-104(BASE), CURVCB		1146
	50		61 D0 00014	MOVL	(CURVCB), FCB		1147
	51		50 D1 00017 1\$:	CMPL	FCB, CURVCB		1152
			3C 13 0001A	BEQL	5\$		
	07	0A	A0 91 0001C	CMPB	10(FCB), #7		1153
			30 12 00020	BNEQ	4\$		
24	A0		53 B1 00022	CMPW	FIDNUM, 36(FCB)		1156
			25 12 00026	BNEQ	3\$		
29	A0		52 91 00028	CMPB	FIDNMX, 41(FCB)		1157
			1F 12 0002C	BNEQ	3\$		
		18	A0 B5 0002E	TSTW	24(FCB)		1158
			06 12 00031	BNEQ	2\$		
		00B0	C0 D5 00033	TSTL	176(FCB)		
			14 13 00037	BEQL	3\$		
22	A0		01 8A 00039 2\$:	BICB2	#1, 34(FCB)		1175
		0B	A0 95 0003D	TSTB	11(FCB)		1183
			18 12 00040	BNEQ	6\$		
		48	A0 D5 00042	TSTL	72(FCB)		1184
			13 13 00045	BEQL	6\$		
23	A0		01 88 00047	BISB2	#1, 35(FCB)		1186
			0D 11 0004B	BRB	6\$		1188
	50		60 D0 0004D 3\$:	MOVL	(FCB), FCB		1191
			C5 11 00050	BRB	1\$		1153
			FEFF 00052 4\$:	BUGW			1194
			0000* 00054	.WORD	<BUG\$_NOTFCBFCB!4>		
			BF 11 00056	BRB	1\$		1153
			50 D4 00058 5\$:	CLRL	FCB		1199
	12		00 DA 0005A 6\$:	MTPR	#0, #18		1202
			04 0005D	RET			1206

SCHFCB
V04-000

H 1
16-Sep-1984 01:08:46
14-Sep-1984 12:30:45

VAX-11 Bliss-32 V4.0-742
DISK\$VMSMASTER:[F11X.SRC]SCHFCB.B32;1 Page 6 (2)

; Routine Size: 94 bytes, Routine Base: \$LOCKEDC1\$ + 0000

: 218 1207 1
: 219 1208 1 END
: 220 1209 0 ELUDOM

PSECT SUMMARY

Name	Bytes	Attributes
\$LOCKEDC1\$	94	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619	33	0	1000	00:02.0

COMMAND QUALIFIERS

; BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:SCHFCB/OBJ=OBJ\$:SCHFCB MSRC\$:SCHFCB/UPDATE=(ENH\$:SCHFCB)

; Size: 94 code + 0 data bytes
; Run Time: 00:16.4
; Elapsed Time: 00:30.6
; Lines/CPU Min: 4436
; Lexemes/CPU-Min: 52223
; Memory Used: 200 pages
; Compilation Complete

0173 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SCHFCB
LIS

SND5MB
LIS

SHFDIR
LIS

SNDER
LIS

TRUNC
LIS

FAL

FAL
MAP

SELVOL
LIS

DAPDEF
MOL

SMALOC
LIS

SNOBAD
LIS

SWTTL
LIS

WITURN
LIS